

EEEEEEEEEEEEEEEE	DDDDDDDDDDDDDD	TTTTTTTTTTTTTTTT
EEEEEEEEEEEEEEEE	DDDDDDDDDDDDDD	TTTTTTTTTTTTTTTT
EEEEEEEEEEEEEEEE	DDDDDDDDDDDDDD	TTTTTTTTTTTTTTTT
EEE	DDD	TTT
EEE	DDD	TTT
EEE	DDD	TTT
EEE	DDD	TTT
EEE	DDD	TTT
EEE	DDD	TTT
EEEEEEEEEEEEEE	DDD	TTT
EEEEEEEEEEEEEE	DDD	TTT
EEEEEEEEEEEEEE	DDD	TTT
EEE	DDD	TTT
EEE	DDD	TTT
EEE	DDD	TTT
EEE	DDD	TTT
EEE	DDD	TTT
EEEEEEEEEEEEEEEE	DDDDDDDDDDDDDD	TTT
EEEEEEEEEEEEEEEE	DDDDDDDDDDDDDD	TTT
EEEEEEEEEEEEEEEE	DDDDDDDDDDDDDD	TTT

:

```
WW      WW      000000      RRRRRRRR      DDDDDDDD      WW      WW      RRRRRRRR      AAAAAA      PPPPPPPP
WW      WW      000000      RRRRRRRR      DDDDDDDD      WW      WW      RRRRRRRR      AAAAAA      PPPPPPPP
WW      WW      00      00      RR      RR      DD      DD      WW      WW      RR      RR      AA      AA      PP      PP
WW      WW      00      00      RR      RR      DD      DD      WW      WW      RR      RR      AA      AA      PP      PP
WW      WW      00      00      RR      RR      DD      DD      WW      WW      RR      RR      AA      AA      PP      PP
WW      WW      00      00      RR      RR      DD      DD      WW      WW      RR      RR      AA      AA      PP      PP
WW      WW      00      00      RRRRRRRR      DD      DD      WW      WW      RRRRRRRR      AA      AA      PPPPPPPP
WW      WW      00      00      RRRRRRRR      DD      DD      WW      WW      RRRRRRRR      AA      AA      PPPPPPPP
WW      WW      00      00      RR      RR      DD      DD      WW      WW      RR      RR      AAAAAAAAAA      PP
WW      WW      00      00      RR      RR      DD      DD      WW      WW      RR      RR      AAAAAAAAAA      PP
WW      WW      00      00      RR      RR      DD      DD      WWW      WWW      RR      RR      AA      AA      PP
WW      WW      00      00      RR      RR      DD      DD      WWW      WWW      RR      RR      AA      AA      PP
WW      WW      00      00      RR      RR      DD      DD      WWW      WWW      RR      RR      AA      AA      PP
WW      WW      000000      RR      RR      DDDDDDDD      WW      WW      RR      RR      AA      AA      PP
WW      WW      000000      RR      RR      DDDDDDDD      WW      WW      RR      RR      AA      AA      PP
```

....  
....  
....  
....

```
LL      IIIIII      SSSSSSSS
LL      IIIIII      SSSSSSSS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SSSSSS
LL      II      SSSSSS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SS
LLLLLLLLLL      IIIIII      SSSSSSSS
LLLLLLLLLL      IIIIII      SSSSSSSS
```

: R

:  
:

```
0001 0 ZTITLE 'EDT$WORDWRAP - do word wrapping'
0002 0 MODULE EDT$WORDWRAP (
0003 0 IDENT = 'V04-000'
0004 0 ) =
0005 1 BEGIN
0006 1
0007 1 *****
0008 1 *
0009 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0010 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0011 1 * ALL RIGHTS RESERVED.
0012 1 *
0013 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0014 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0015 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0016 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0017 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0018 1 * TRANSFERRED.
0019 1 *
0020 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0021 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0022 1 * CORPORATION.
0023 1 *
0024 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0025 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0026 1 *
0027 1 *
0028 1 *****
0029 1
0030 1
0031 1 ++
0032 1 FACILITY: EDT -- The DEC Standard Editor
0033 1
0034 1 ABSTRACT:
0035 1
0036 1 This module trys to do word wrapping after an insert.
0037 1
0038 1 ENVIRONMENT: Runs at any access mode - AST reentrant
0039 1
0040 1 AUTHOR: John Sauter, CREATION DATE: April 7, 1982
0041 1
0042 1 MODIFIED BY:
0043 1
0044 1 1-001 - Original. JBS 07-Apr-1982
0045 1 1-002 - New screen update logic. JBS 13-Sep-1982
0046 1 1-003 - Add a parameter to split line routine. SMB 16-Nov-1982
0047 1 1-004 - Change the parameter to the split line routine. SMB 17-Nov-1982
0048 1 --
0049 1
```



EDT\$WORDWRAP  
V04-000

EDT\$WORDWRAP - do word wrapping  
Declarations

M 1  
16-Sep-1984 02:17:18  
14-Sep-1984 12:25:49

VAX-11 Bliss-32 V4.0-742  
DISK\$VMMASTER:[EDT.SRC]WORDWRAP.BLI;1  
Page 2  
(2)

```
.. 51      0050 1 %SBTTL 'Declarations'
.. 52      0051 1
.. 53      0052 1 : TABLE OF CONTENTS:
.. 54      0053 1 :
.. 55      0054 1
.. 56      0055 1 REQUIRE 'EDTSRC:TRAROUNAM';
.. 57      0494 1
.. 58      0495 1 FORWARD ROUTINE
.. 59      0496 1     EDT$WORD_WRAP;
.. 60      0497 1
.. 61      0498 1 :
.. 62      0499 1 : INCLUDE FILES:
.. 63      0500 1 :
.. 64      0501 1
.. 65      0502 1 REQUIRE 'EDTSRC:EDTREQ';
.. 66      0637 1
.. 67      0638 1 :
.. 68      0639 1 : MACROS:
.. 69      0640 1 :
.. 70      0641 1 :     NONE
.. 71      0642 1 :
.. 72      0643 1 : EQUATED SYMBOLS:
.. 73      0644 1 :
.. 74      0645 1 :     NONE
.. 75      0646 1 :
.. 76      0647 1 : OWN STORAGE:
.. 77      0648 1 :
.. 78      0649 1 :     NONE
.. 79      0650 1 :
.. 80      0651 1 : EXTERNAL REFERENCES:
.. 81      0652 1 :
.. 82      0653 1 :     In the routine
```

\*\*F

```
84 0654 1 XSBTTL 'EDT$$WORD_WRAP - do word wrapping'
85 0655 1
86 0656 1 GLOBAL ROUTINE EDT$$WORD_WRAP          ! Do word wrapping
87 0657 1 =
88 0658 1
89 0659 1 ++
90 0660 1 FUNCTIONAL DESCRIPTION:
91 0661 1
92 0662 1     This routine is called after text has been inserted to do word wrapping
93 0663 1     if any is called for.
94 0664 1
95 0665 1 FORMAL PARAMETERS:
96 0666 1
97 0667 1     NONE
98 0668 1
99 0669 1 IMPLICIT INPUTS:
100 0670 1
101 0671 1     EDT$$T_LN_BUF
102 0672 1     EDT$$A_LN_PTR
103 0673 1     EDT$$A_LN_END
104 0674 1     EDT$$G_WD_WRAP
105 0675 1
106 0676 1 IMPLICIT OUTPUTS:
107 0677 1
108 0678 1     NONE
109 0679 1
110 0680 1 ROUTINE VALUE:
111 0681 1
112 0682 1     Always 1.
113 0683 1
114 0684 1 SIDE EFFECTS:
115 0685 1
116 0686 1     NONE
117 0687 1
118 0688 1 --
119 0689 1
120 0690 2 BEGIN
121 0691 2
122 0692 2 EXTERNAL ROUTINE
123 0693 2     EDT$$FND_CHWID,
124 0694 2     EDT$$CS_LEFT,
125 0695 2     EDT$$CS_RIGHT,
126 0696 2     EDT$$FND_BWD,
127 0697 2     EDT$$SPLT_LNINS;
128 0698 2
129 0699 2 EXTERNAL
130 0700 2     EDT$$T_LN_BUF,          ! Current line buffer.
131 0701 2     EDT$$A_LN_PTR,          ! Current character pointer.
132 0702 2     EDT$$A_LN_END,          ! End of current line pointer.
133 0703 2     EDT$$G_WD_WRAP;          ! Word wrap point.
134 0704 2
135 0705 2 LOCAL
136 0706 2     CP,
137 0707 2     SAVE_POINT,
138 0708 2     DIFF,
139 0709 2     COL;
140 0710 2
```

```
141 0711 2 !+
142 0712 2 !- Do nothing if there is no word wrapping.
143 0713 2 !-
144 0714 2
145 0715 2 IF (.EDT$$G_WD_WRAP NEQ 256)
146 0716 2 THEN
147 0717 2 BEGIN
148 0718 2 !+
149 0719 2 !- Compute the current cursor position.
150 0720 2
151 0721 2 COL = 0;
152 0722 2 CP = CH$PTR (EDT$$T_LN_BUF);
153 0723 2
154 0724 2 WHILE CH$PTR NEQ (.CP, .EDT$$A_LN_PTR) DO
155 0725 2 COL = .COL + EDT$$FMT_CHWID (CH$RCHAR_A (CP), .COL);
156 0726 2
157 0727 2 !+
158 0728 2 !- If the current column is beyond the wrap point, wrap the line.
159 0729 2
160 0730 2
161 0731 2 IF (.COL GTR .EDT$$G_WD_WRAP)
162 0732 2 THEN
163 0733 2 BEGIN
164 0734 2 SAVE_POINT = .EDT$$A_LN_PTR;
165 0735 2 EDT$$CS_LEFT ();
166 0736 2
167 0737 2 IF (CH$RCHAR (.EDT$$A_LN_PTR) EQL %C' ') THEN EDT$$CS_RIGHT () ELSE EDT$$FND_BWD (1);
168 0738 2
169 0739 2 DIFF = CH$DIFF (.SAVE_POINT, .EDT$$A_LN_PTR);
170 0740 2
171 0741 2 IF CH$PTR_NEQ (.EDT$$A_LN_PTR, CH$PTR (EDT$$T_LN_BUF)) THEN EDT$$SPLT_LNINS (1); ! Use optimi
172 0742 2
173 0743 2 EDT$$A_LN_PTR = CH$PLUS (.EDT$$A_LN_PTR, .DIFF);
174 0744 2 END;
175 0745 2
176 0746 2 END;
177 0747 2 RETURN (1);
178 0748 2
179 0749 1 END;
```

! of routine EDT\$\$WORD\_WRAP

```
.TITLE EDT$WORDWRAP EDT$WORDWRAP - do word wrapping
.IDENT \V04-000\
```

```
.EXTRN EDT$$FMT_CHWID, EDT$$CS_LEFT
.EXTRN EDT$$CS_RIGHT, EDT$$FND_BWD
.EXTRN EDT$$SPLT_LNINS
.EXTRN EDT$$T_LN_BUF, EDT$$A_LN_PTR
.EXTRN EDT$$A_LN_END, EDT$$G_WD_WRAP
```

```
.PSECT _EDT$CODE NOWRT, SHR, PIC,2
```

```
.ENTRY EDT$$WORD_WRAP, Save R2,R3,R4,R5,R6 : 0656
MOVAB EDT$$G_WD_WRAP, R6 :
MOVAB EDT$$T_LN_BUF, R5 :
MOVAB EDT$$A_LN_PTR, R4 :
CMPL EDT$$G_WD_WRAP, #256 : 0715
```

```
00000100 56 00000000G 00 007C 00000
55 00000000G 00 9E 00002
54 00000000G 00 9E 00009
BF 00000000G 00 9E 00010
66 D1 00017
```



EDT\$WORDWRAP  
V04-000

EDT\$WORDWRAP - do word wrapping  
EDT\$\$WORD\_WRAP - do word wrapping

C 2  
16-Sep-1984 02:17:18  
14-Sep-1984 12:25:49

VAX-11 Bliss-32 V4.0-742  
DISK\$VMSMASTER:[EDT.SRC]WORDWRAP.BLI;1  
Page 5  
(3)

	5E	13	0001E	BEQL	6\$		
	53	D4	00020	CLRL	COL		0721
52	65	9E	00022	MOVAB	EDT\$ST_LN_BUF, CP		0722
64	52	D1	00025	1\$: CMPL	CP, EDT\$SA_LN_PTR		0724
	11	13	00028	BEQL	2\$		
	53	DD	0002A	PUSHL	COL		0725
7E	82	9A	0002C	MOVZBL	(CP)+, -(SP)		
00000000G	00	02	FB	0002F	CALLS	#2, EDT\$FMT_CHWID	
	53	5C	C0	00036	ADDL2	R0, COL	
		EA	11	00039	BRB	1\$	
66	53	D1	0003B	2\$: CMPL	COL, EDT\$G_WD_WRAP		0731
	3E	15	0003E	BLEQ	6\$		
52	64	D0	00040	MOVL	EDT\$SA_LN_PTR, SAVE_POINT		0734
00000000G	00	00	FB	00043	CALLS	#0, EDT\$SCS_LEFT	0735
	50	64	D0	0004A	MOVL	EDT\$SA_LN_PTR, R0	0737
20	60	91	0004D	CMPB	(R0), #32		
	09	12	00050	BNEQ	3\$		
00000000G	00	00	FB	00052	CALLS	#0, EDT\$SCS_RIGHT	
	09	11	00059	BRB	4\$		
00000000G	00	01	DD	0005B	3\$: PUSHL	#1	
	51	64	D0	0005D	CALLS	#1, EDT\$FND_BWD	
52	51	C2	00067	4\$: MOVL	EDT\$SA_LN_PTR, R1		0739
50	65	9E	0006A	SUBL2	R1, DIFF		
50	51	D1	0006D	MOVAB	EDT\$ST_LN_BUF, R0		0741
	09	13	00070	CMPL	R1, R0		
	01	DD	00072	BEQL	5\$		
00000000G	00	01	FB	00074	PUSHL	#1	
	64	52	C0	0007B	CALLS	#1, EDT\$SPLT_LNINS	
50	01	D0	0007E	5\$: ADDL2	DIFF, EDT\$SA_LN_PTR		0743
		04	00081	6\$: MOVL	#1, R0		0748
				RET			0749

; Routine Size: 130 bytes, Routine Base: \_EDT\$CODE + 0000

; 180 0750 1  
; 181 0751 1 !<BLF/PAGE>

EDT\$WORDWRAP  
V04-000

EDT\$WORDWRAP - do word wrapping  
EDT\$\$WORD\_WRAP - do word wrapping

D 2  
16-Sep-1984 02:17:18  
14-Sep-1984 12:25:49

VAX-11 Bliss-32 V4.0-742  
DISK\$VMSMASTER:[EDT.SRC]WORDWRAP.BLI;1 (4) Page 6

: 183 0752 1 END  
: 184 0753 1  
: 185 0754 0 ELUDOM

! of module EDT\$WORDWRAP

# PSECT SUMMARY

Name	Bytes	Attributes
_EDT\$CODE	130	NOVEC,NOWRT, RD , EXE, SHR, LCL, REL, CON, PIC,ALIGN(2)

# Library Statistics

File	----- Total	Symbols Loaded	----- Percent	Pages Mapped	Processing Time
\$255\$DUA28:[EDT.SRC]EDT.L32;1	377	1	0	40	00:00.2
\$255\$DUA28:[EDT.SRC]PSECTS.L32;1	2	1	50	7	00:00.1

# COMMAND QUALIFIERS

BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/NOTRACEBACK/LIS=LIS\$:WORDWRAP/OBJ=OBJ\$:WORDWRAP MSRC\$:WORDWRAP.BLI/UPDATE=(ENH\$:WORDWRAP)

: Size: 130 code + 0 data bytes  
: Run Time: 00:11.6  
: Elapsed Time: 00:14.9  
: Lines/CPU Min: 3910  
: Lexemes/CPU-Min: 10983  
: Memory Used: 74 pages  
: Compilation Complete



0142 AH-BT13A-SE  
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION  
CONFIDENTIAL AND PROPRIETARY

